Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Departement of Computer Science
Markus Püschel, David Steurer
Johannes Lengler, Gleb Novikov, Chris Wendler, Ulysse Schaller

19. October 2020

Algorithms & Data Structures — Exercise sheet 5 — HS 20

Exercise Class (Room & TA): ______________________

Submitted by: ______________________

Peer Feedback by: ______________________

Points: _____

Submission: On Monday, 26 October 2020, hand in your solution to your TA *before* the exercise class starts. Exercises that are marked by * are challenge exercises. They do not count towards bonus points.

Exercise 5.1 *Heapsort* (1 point).

Given the array [H, E, A, P, S, O, R, T], we want to sort it in ascending alphabetical order using Heapsort.

a) Draw the interpretation of the array as a heap, before any call of RestoreHeapCondition.

b) In the lecture you have learned a method to construct a heap from an unsorted array (see also pages 35–36 in the script). Draw the resulting max binary heap if this method is applied to the above array.

c) Sort the above array in ascending alphabetical order with heapsort, beginning with the heap that you obtained in (b). Draw the array after each intermediate step in which a key is moved to its final position.

Exercise 5.2 *Sorting algorithms* ***(This exercise is from Summer 2020 exam)*** .

Below you see four sequences of snapshots, each obtained during the execution of one of the following algorithms: `InsertionSort`, `SelectionSort`, `QuickSort`, `MergeSort`, and `BubbleSort`. For each sequence, write down the corresponding algorithm.

8	6	4	2	5	1	3	7
6	4	2	5	1	3	7	8
4	2	5	1	3	6	7	8

____

8	6	4	2	5	1	3	7
1	6	4	2	5	8	3	7
1	2	4	6	5	8	3	7

____

8	6	4	2	5	1	3	7
6	8	2	4	1	5	3	7
2	4	6	8	1	3	5	7

____

8	6	4	2	5	1	3	7
6	8	4	2	5	1	3	7
4	6	8	2	5	1	3	7

____

Exercise 5.3 *Counting Operations in Loops II.*

For the following code fragments count how many times the function f is called. Report the number of calls as nested sum, and then simplify your expression in simplified Θ-notation and prove your result.

Hint: *Note that in order to justify your Θ-notation you are required to show two parts: an upper bound on your nested sum as well as a lower bound.*

a) Consider the snippet:

Algorithm 1

```
for j = 1, ..., n do
    k ← 1
    while k ≤ j do
        m ← 1
        while m ≤ j do
            f()
            m ← 2 · m
        k ← 2 · k
```

b) Consider the snippet:

Algorithm 2

```
for j = 1, ..., n do
    for l = 1, ..., 100 do
        k ← 1
        while k^2 ≤ j do
            f()
            f()
            k ← k + 1
```

Exercise 5.4 *Mastermind* **(1 point)**.

Anna and Ben are playing *Mastermind*. The game consists of pins of 6 different colors taken from the set $C = \{1, 2, 3, 4, 5, 6\}$. Anna secretly chooses a combination of four of these pins (not necessarily of different colors), i.e. a tuple $a = (a_1, a_2, a_3, a_4) \in C^4$. Ben's goal is to discover this tuple. For every guess $b = (b_1, b_2, b_3, b_4) \in C^4$ that Ben does, Anna tells him how many correct pins there are in b, i.e. for how many indices $i \in \{1, 2, 3, 4\}$ one has $a_i = b_i$. For example, if $a = (2, 1, 2, 1)$ and $b = (5, 1, 1, 1)$, Anna tells Ben that his guess contains 2 correct pins (but she does not tell him which positions are correct).

a) How many different combinations of pins can Anna make ?

b) Suppose that, after a few guesses, Ben has reduced to k the number of possible combinations (that is, the number of tuples a that are compatible with all of Anna's answers). Show that it is impossible for Ben to reduce *for sure* the number of possible combinations to strictly less than $\lceil \frac{k-1}{4} \rceil$ with his next guess.

   Hint: *How many possible answers can Anna give to the Ben's next guess? How many combinations correspond to the the answer that the Ben's guess contains* 4 *correct pins?*

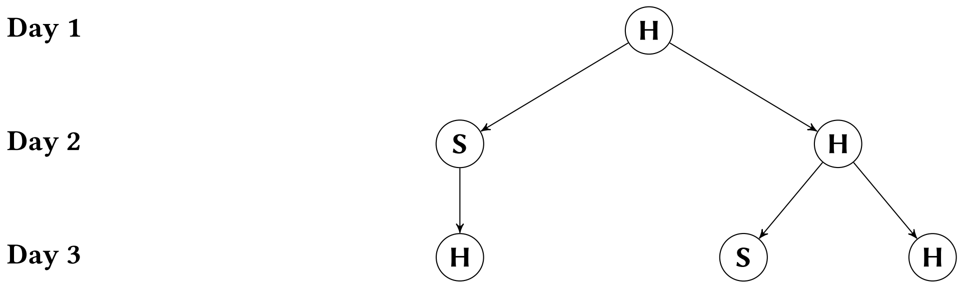
c) Use parts (a) and (b) to show that, for any strategy that Ben uses to make his guesses, there always exists a tuple $a \in C^4$ that Ben cannot attain in strictly less than 6 guesses. In other words, there is no algorithm that Ben can implement which has a worst-case runtime (in terms of number of guesses) strictly less than 6.

Exercise 5.5 *Wine tasting* **(1 point)**.

You have n barrels of different wines and you want to organize a wine tasting event. However, you learn that exactly one of the wine barrels is poisoned. Your friend Céline is really into wine and agrees to taste the wines in advance in order to help you find the poisoned barrel. On the i-th day, Céline can try as many different wines as you want. However, the effect of the poisoning is slow, so you only learn on the morning of day $i+1$ whether one of the drunken wines was poisoned: if yes, she is sick for the whole day and cannot drink any wine during this day, but she will be fully recovered for day $i+2$; if not, she can again drink as many wines on day $i+1$ as she wants. In any case, you learn whether one of the wines from the i-th day was poisoned, but you don't know which one it might be. You want to know how many days in the worst case you will need to discover which barrel is poisoned.

a) Draw a decision tree that shows how many different cases (where one case corresponds to one specific barrel being poisoned) you can distinguish after k days, for $k = 4$ (in other words your tree should have depth 4).

   Below you can see the decision tree which corresponds to the case $k = 3$:



   An **S** node indicates that Céline is sick on that day, i.e. she is not able to drink any wine, while an **H** node indicates that Céline is healthy on that day and can try some wines. The leaves of this tree correspond to 3 different cases which is possible to distinguish in 3 days.

b) Denote by C_k the number of cases that you can distinguish after k days. Find a recurrence relation of the form $C_k = f(C_{k-1}, C_{k-2})$ and justify why it holds. The decision tree from part (a) could be useful.

   Hint: *C_k corresponds to the number of leaves at depth k in the decision tree.*

c) Use the recurrence relation found in part (b) to show by induction that $C_k \leq 2^k$.

*c') Alternatively to (c), let $\varphi := \frac{1+\sqrt{5}}{2}$, and show that $C_k \leq \varphi^k$.

d) Deduce that, no matter what the drinking strategy of Céline is, you will need at least $\Omega(\log n)$ days to find the poisoned wine in the worst case.

e) Is there a strategy that is guaranteed to succeed in $\mathcal{O}(\log n)$ days? Justify your answer.